Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Miha Novak

Pameten protivlomen sistem

diplomsko delo

univerzitetni študijski program
prve stopnje
računalništvo in informatika

Mentor: izr. prof. dr. Patricio Bulić

Ljubljana, 2019

Besedilo je oblikovano z urejevalnikom besedil LaTeX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Vlomi v stanovanja se največkrat zgodijo, ko so stanovanja prazna. Zaradi tega implementirajte rešitev s katero boste stanovanje spremenili v pametno in postavili pameten protivlomen varnostni sistem. Identificirajte šibke točke stanovanja ter postavite zahteve, ki jih mora pameten protivlomen varnostni sistem izpolnjevati in temu primerno izberite pametne naprave (senzorje in aktuatorje). Po postavitvi protivlomnega varnostnega sistema zaženite testne primere s katerimi boste sistem analizirali in ovrednotili.

Zahvaljujem se Maji in svoji družini za podporo in potrpežljivost v času študija. Hvala!

Kazalo

Seznam uporabljenih kratic

kratica	angleško	slovensko
IoT	Internet of Things	internet stvari
HA	Home Assistant	
RPI	Raspberry PI	
IP	Internet Protocol	internetni protokol
DNS	Domain Name System	sistem domenskih imen
API	Application programming interface	aplikacijski programski vmesnik

Povzetek

Naslov: Pameten protivlomen sistem

Avtor: Miha Novak

Statistika slovenske policije pravi, da se je v letu 2017 zgodilo nekaj več kot 37.000, v prvi polovici leta 2018 pa nekaj več kot 20.000 kaznivih dejanj zoper premoženja. V to skupino kaznivih dejanj spadajo tudi vlomi v stanovanja, ki se po statističnih podatkih največkrat zgodijo, ko so stanovanja prazna. Zaradi tega dejstva smo se odločili, da bomo postavili pameten brezžičen protivlomen sistem. Identificirali smo šibke točke našega stanovanja, postavili zahteve, ki jih mora naš protivlomen sistem izpolnjevati in temu primerno izbrali pametne naprave (senzorje in aktuatorje), ki smo jih namestili in povezali s centralno kontrolno enoto. Na centralno enoto smo namestili odprtokodno platformo za upravljanje pametnega doma, preko katere bomo lahko upravljali naš protivlomen sistem. Po postavitvi protivlomnega sistema, smo poskrbeli tudi za varen dostop do sistema preko interneta in glede na naše zahteve postavili tudi testne primere, po katerih smo protivlomen sistem tudi testirali.

Ključne besede: internet stvari, pameten protivlomen sistem, pametne naprave, senzor, aktuator, varnost.

Abstract

Title: A smart anti-theft system

Author: Miha Novak

Slovenian police statistics says that in year 2017 happened more than 37,000 crimes against property and in the first half of 2018 something more than 20,000. This crimes also includes home burglaries. Statistically most of the burglaries occur when the homes are empty. Because of this we decided to set up a smart wireless anti-theft system. We identified the weak points of our home and create requirements that our smart anti-theft system must fulfill. According to the requirements we chose smart devices (sensors and actuators) which we installed and connected to the central control unit. On central control unit we installed open source platform for managing smart home. After the installation and configuration of the anti-theft system, we also provided secure access to the system over the Internet. According to our requirements, we set up test cases, according to which our wireless anti-theft system has been tested.

Keywords: Internet of Things, a smart anti-theft-system, smart devices, sensor, actuator, security.

Poglavje 1

Uvod

Število aktivnih pametnih naprav je v letu 2018 znašalo skoraj 7 miljard. Napovedi kažejo, da naj bi število aktivnih pametnih naprav do leta 2025 naraslo na več kot 21,5 miljard[1] [15]. Pametne naprave se povezujejo v pametna omrežja. Končni uporabniki se s pojmom pametnih omrežij najpogosteje srečamo v povezavi s pametnim domom (ang. smart home). Spletni iskalnik Google vrne ob iskanem pojmu *smart home* (slo. pameten dom) skoraj 4 miljarde zadetkov.

Pameten dom nam omogoča oddaljeno upravljanje nameščenih pametnih naprav (npr. vklapljanje in izklapljanje luči), omogoča nam nadzor nad porabo energije in s tem zmanjševanje stroškov, omogoča nam varnostni nadzor nad domom in še mnogo drugih stvari.

Cilj našega diplomskega dela je predstaviti internet stvari in postaviti pameten brezžičen protivlomen varnostni sistem. Varnostni sistem bomo postavili in nastavili tako, da bo komuniciral s centralno kontrolno enoto na kateri bo nameščena odprtokodna programska oprema, ki omogoča avtomatizacijo in kontrolo nad pametnimi napravami nameščenimi v našem domu.

[1] v število naprav niso vključeni pametni telefoni, tablice, prenosni računalniki in stacionarni telefoni

Poglavje 2

Internet stvari

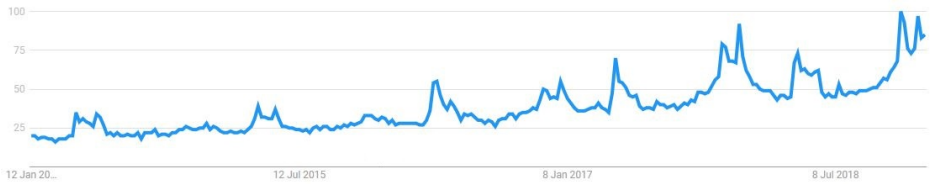
Internet stvari (ang. Internet of Things) je vrsta omrežja katerihkoli fizičnih objektov, stvari (ang. things), ki med seboj na podlagi določenih protokolov, komunicirajo in delijo informacije. To je omrežje v katerem niso povezani samo računalniki ampak tudi vse ostale naprave različnih tipov in velikosti (npr. pametni telefoni, senzorji, aktuatorji, vozila, gospodinjski aparati, zabavna elektronika) [20].

Pojem internet stvari, za katerega bomo uporabili splošno znano kratico IoT, se je prvič pojavil leta 1999. Prva implementacija koncepta pa sega v leto 1982, ko so raziskovalci iz univerze Carnegie Mellon uspeli povezati kokakolin avtomat za pijačo z internetom. Implementacija je omogočala uporabniku vpogled v zalogo pijače in ali je pijača primerne temperature za pitje [8, 14].

Danes se pametne naprave, ki se povezujejo v internet uporabljajo tako v domačem okolju, za osebno rabo, kot tudi v industriji. Pametne naprave, ki jih vsakodnevno uporabljamo so pametni telefoni, pametne ure ali pa pametne naprave, ki naše domovanje spremenijo v pametno (ang. smart home) in nam omogočajo lagodnejše in brezskrbno bivanje. Primeri naprav, ki jih lahko namestimo v domove so pametne vtične, ki merijo porabo električne energije, pametne ključavnice, ki nam omogočajo oddaljeno odklepanje in zaklepanje vrat, pametna stikala, ki jih lahko oddaljeno preklapljamo, pametni varnostni sistemi in tako naprej. Nič drugače ni v industriji, kjer s

pomočjo pametnih naprav (senzorjev, aktuatorjev) povezanih v omrežje IoT, zbirajo ogromne količine podatkov, ki jih analizirajo in glede na njih optimizirajo poslovne procese. Take vrste omrežja najdemo na različnih področjih kot so zdravstvo, komunikacije, finančne institucije, proizvodnja, prodaja, energetika, prevozništvo, kmetijstvo, itd. [10].

Število aktivnih IoT naprav je od leta 2015 naraslo za več kot 50% in je v letu 2018 znašalo skoraj 7 miljard. Kot kažejo napovedi, se bo trend rasti nadaljeval, saj naj bi število aktivnih naprav do leta 2025 naraslo na več kot 21,5 miljard[1]. Leta 2025 naj bi tržna vrednost IoT presegla 1567 miljard ameriških dolarjev. Za primerjavo naj povemo, da je v letu 2018 znašala 151 miljard ameriških dolarjev [15]. Trend rasti in zanimanja lahko vidimo tudi na grafu iskanj v spletnem iskalniku Google za iskani izraz *smart home* (slo. pameten dom) (graf 2.1). Graf predstavlja zanimanje za iskan izraz glede na najvišjo točko za dano regijo (izbrali smo vsa svetovna iskanja) in čas (izbrali smo vsa iskanja od leta 2013). Vrednost 100 predstavlja največjo priljubljenost izraza, vrednost 50 pomeni, da je izraz pol manj priljubljen, vrednost 0 pa, da za ta izraz ni bilo iskanj.



Slika 2.1: Trend iskanja v spletnem iskalniku Google za izraz *smart home*

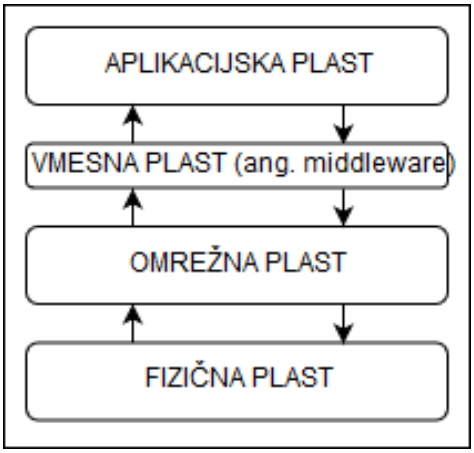
[1] v število naprav niso vključeni pametni telefoni, tablice, prenosni računalniki in stacionarni telefoni

2.1 Poenostavljen arhitekturni model IoT

V zadnjih nekaj letih je bilo objavljenih več referenčnih arhitekturnih modelov IoT, ki jim je skupno to, da zajemajo celoten tok podatkov, od generiranja podatkov na pametnih napravah, procesiranja podatkov, prenosa podatkov in do končne obdelave podatkov v posameznih aplikacijah [24].

Namen arhitekturnega referenčnega modela je, da zagotovi nedvoumno definicijo in opis arhitekture, ki se jo lahko implementira v praksi. Primer arhitekturnega referenčnega modela IoT je sedem plastni arhitekturni referenčni model, ki ga je leta 2014 predstavil arhitekturni odbor, ki so ga vodili Cisco, IBM, Rockwell Automation in ostali [12].

Kljub temu, da se referenčni arhitekturni modeli IoT med seboj razlikujejo in so odvisni od področja za katerega so definirani, jim je skupno to, da definirajo smernice implementacije IoT za pametne naprave, ki si preko komunikacijskega omrežja izmenjujejo podatke, ki se uporabijo v aplikacijah. IoT arhitekturo lahko poenostavljeno predstavimo s tremi plastmi: fizično plastjo, omrežno plastjo in aplikacijsko plastjo. Vsako od plasti, ki so prikazane na sliki 2.2, bomo opisali v naslednjih podpoglavjih.



Slika 2.2: Poenostavljen arhitekturni model IoT

2.1.1 Pametne naprave

Pametne naprave, stvari (ang. things) v IoT, so vse naprave, ki so zmožne zajemanja podatkov, pošiljanja podatkov, prejemanja podatkov in pretvarjanja podatkov iz analogne v digitalno obliko. Primera takih naprav sta senzor in aktuator.

Senzor je naprava, ki proizvede na izhodu signal (npr. električno napetost), ki enolično odgovarja vrednosti opazovane veličine na vhodu in ga s pomočjo pripadajoče elektronike pretvori v električni ali digitalen signal, ki je primeren za nadaljno obdelavo [3]. Za primer lahko vzamemo senzor zvoka, ki na vhodu zaznava razliko v zračnem tlaku in na izhodu proizvede električen signal, ki enolično odgovarja spremembi zračnega tlaka. Na podlagi zazanave senzorja, lahko zvok predvajamo preko zvočnika (aktuator).

Aktuator oz. vzbujevalnik je element, ki vhodni signal pretvarja v mehanski ali informacijski izhodni signal [3]. Aktuator sprejme vhodni signal, ki je lahko električen ali digitalen in glede na signal sproži fizično akcijo kot je predvajanje zvoka.

2.1.2 Fizična plast

Na fizični plasti se nahajajo vse pametne naprave. Glavna naloga fizične plasti je zaznavanje (ang. perception) fizičnih lastnosti stvari okoli nas. Poleg tega fizična plast skrbi tudi za zbiranje in pripravo podatkov za prenos preko omrežne plasti [32]. Na fizični plasti se vsi podatki pretvorijo iz analognega v digitalen signal, ki je bolj primeren za prenos po omrežju.

2.1.3 Omrežna plast

Omrežna plast poenostavljenga arhitekturnega modela skrbi za procesiranje podatkov prejetih iz fizične plasti in skrbi za prenos podatkov do aplikacijske plasti [1]. Prenos podatkov se vrši skozi različne vrste omrežji npr. brezžična omrežja, podatki pa se prenašajo preko različnih tehnologij npr. WiFi, Bluetooth, ZigBee.

Zaradi velike količine podatkov, ki se prenašajo iz fizične plasti preko omrežne plasti do aplikacijske plasti, potrebujemo tudi vmesno plast (ang. middleware) na kateri se bodo podatki shranjevali in procesirali. Tipično se za to uporablja oblakovno računalništvo (ang. cloud computing). To je izraz, ki označuje uporabo oddaljenih strežnikov, namesto lokalnih strežnikov oziroma osebnih računalnikov, na katerih se podatki shranjujejo, obdelujejo in procesirajo.

2.1.4 Aplikacijska plast

Aplikacijska plast obdelane podatke iz omrežne plasti interpretira in pripravi za uporabo v aplikacijah. Na tej plasti se lahko nahajajo raznovrstne aplikacije kot so aplikacije za poročanje, obdelavo in analizo podatkov in aplikacije za nadzor.

2.1.5 Komunikacijski protokoli in tehnologije

Pametne naprave med seboj komunicirajo v skladu s komunikacijskimi protokoli. Komunikacijski protokol je definiran kot skupek pravil, ki omogočajo dvema ali večim napravam v omrežju pošiljanje in sprejemanje podatkov po kakršnemkoli fizičnem mediju. Komunikacijski protokoli so lahko implementirani na nivoju strojne opreme, programske opreme ali pa so kombinacija obeh implementacij [26]. V našem pametnem omrežju naprave med seboj komunicirajo preko HTTP protokola (ang. Hypertext Transfer Protocol) [27] in preko brezžičnih tehnologij Bluetooth [25], Zigbee [30] in Wi-Fi [29].

2.2 Primeri uporabe IoT v praksi

2.2.1 Pametna mesta

Primer uporabe IoT v industriji so pametna mesta (ang. smart city). V pametnem mestu se z uporabo IoT izboljša in optimizira delovanje javnih služb, izboljša pa se tudi kvaliteta življenja prebivalcev na tem področju. IoT

omogoča optimizacijo ravnanja z odpadki, nadzor nad kvaliteto zraka, hrupa in prometa (npr. obveščanje o prometnih zastojih, pametno parkiranje) in optimizacijo porabe energije v mestu (npr. pametna osvetljava). Dokaz koncepta (ang. proof of concept) so naredili raziskovalci univerze v Padovi, ki so postavili sistem za spremljanje javne razsvetljave in zbiranje podatkov iz okolja (onesnaženost zraka, temperaturo in vlažnost). S pomočjo zbranih podatkov so lahko na podlagi onesnaženosti zraka ugotovili kdaj je bilo v mestu največ prometa [33].

2.2.2 Pameten dom

Pameten dom (ang. smart home) je eden izmed primerov uporabe IoT pri domačih uporabnikih. Preden lahko definiramo kaj pomeni pameten dom (ang. smart home), moramo najprej predstaviti koncept doma. Koncept doma lahko predstavimo s sledečimi vidiki:

- Dom je prostor kjer se počutimo varne in imamo nadzor.
- Dom je prostor kjer izvajamo vsakodnevne aktivnosti.
- Dom je prostor kjer sobivamo z ljudmi s katerimi smo povezani.
- Dom je prostor, ki nam določa identiteto in nam daje vrednost.

V našem delu se bomo osredotočili na vidik varnosti in nadzora v pametnem domu. Definicij pametnega doma je več. Ena izmed njih pravi, da je pameten dom tisti v katerem so senzorji, upravljalniki in ostale naprave povezane v komunikacijsko omrežje, ki omogoča uporabnikom doma spremljanje in upravljanje teh naprav. [11].

Poglavje 3

Opis problema

Po podatkih slovenske policije se je v letu 2017 zgodilo 37.429, v prvem polletju leta 2018 pa 20.270 kaznivih dejanj zoper premoženja. V to kategorijo spadajo sledeča kazniva dejanja: poškodovanje tuje stvari, vlom, drzna tatvina, rop, roparska tatvina, zatajitev, klasična goljufija, požig in tako dalje. Od tega je bilo leta 2017 preiskanih 28,9% kaznivih dejanj zoper premoženja, v prvem polletju leta 2018 pa 34% [21]. Največ vlomov se zgodi med 10. in 11. uro dopoldne in med 13. in 15. uro popoldne, ko so domovi prazni [2]. Zaradi tega dejstva smo se odločili, da bomo naše stanovanje zavarovali s pametnim protivlomnim varnostnim sistemom.

3.1 Analiza šibkih točk

Prvi korak pred postavitvijo protivlomnega varnostnega sistema je analiza šibkih točk stanovanja. Šibka točka smo poimenovali vse izpostavljene točke (slabosti) preko katerih bi lahko nepridipravi vlomili v naše stanovaje.

Stanovanje je polkletno in veliko približno 50 m^2. V stanovanje se vstopi skozi protivlomna vhodna vrata, ki vodijo na hodnik. Na levi strani hodnika sta spalnica in kopalnica na koncu hodnika je kuhinja z jedilnico in desno od kuhinje z jedilnico, dnevna soba.

Nepridipravi bi lahko v stanovanje vlomili skozi vrata ali pa skozi okno,

ki se nahaja v vsakem prostoru. Zaradi dejstva, da je stanovanje polkletno smo torej poleg vrat tudi vsa okna identificirali kot šibko točko.

3.2 Zahteve protivlomnega varnostnega sistema

Po identifikaciji šibkih točk našega stanovanja smo se odločili, da bomo za ponudbo protivlomnega varnostnega sistema povprašali enega izmed vodilnih slovenskih podjetji, ki se ukvarjajo z varovanjem. V spodnjem seznamu so navedene naše zahteve za protivlomen varnostni sistem:

- Protivlomen varnostni sistem mora biti brezžičen.
- Protivlomen varnostni sistem mora zaznati odpiranje vrat in oken.
- Protivlomen varnostni sistem mora zaznati gibanje v stanovanju.
- Protivlomen varnostni sistem mora ob zaznavi odpiranja vrat in oken vklopiti sireno.
- Protivlomen varnostni sistem mora ob zaznavi gibanja v stanovanju vklopiti sireno.
- Protivlomen varnostni sistem nam mora ob zaznavi odpiranja vrat in oken posredovati sporočilo o dogodku.
- Protivlomen varnostni sistem nam mora ob zaznavi gibanja v stanovanju posredovati sporočilo o dogodku.
- Protivlomen varnostni sistem nam mora omogočati oddaljen dostop (oddaljen vklop in izklop sistema).
- Protivlomen varnostni sistem mora omogočati nastavitev pravil za vklop in izklop sistema.

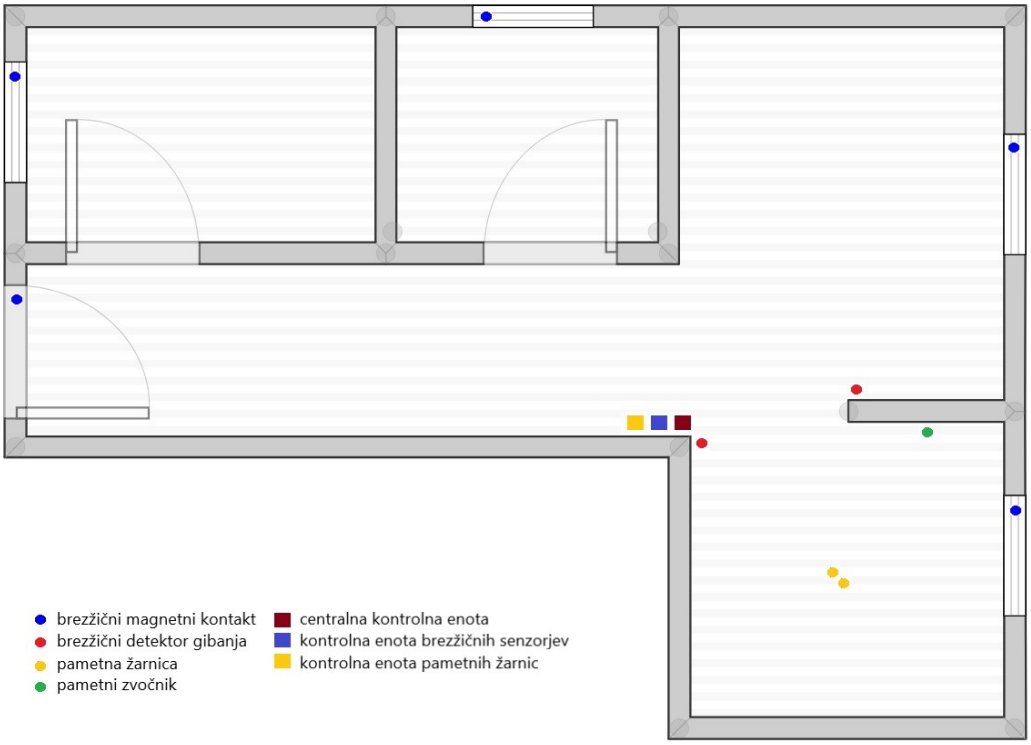
- Protivlomen varnostni sistem se mora samodejno vklopiti, če zazna da je stanovanje prazno.

Ponudba, ki smo jo dobili od podjetja za varovanje je bila korektna ampak žal ni vključevala vseh naših želja. Ponudili so nam brezžičen protivlomen varnostni sistem z brezžičnimi detektorji gibanaja in brezžičnimi magnetnimi kontakti. Sistem, ki so nam ga ponudili ne zna prepoznati, če je stanovanje prazno in se samodejno vklopiti. Dodatna težava bi bila tudi integracija ostalih pametnih naprav s ponujenim sistemom. Zarad dejstva, da bomo postopoma celoten dom spremenili v pametnega in da bi poleg protivlomnega varnostnega sistema radi imeli tudi nadzor nad ostalimi napravami iz enotnega uporabniškega vmesnika, smo se odločili, da bomo protivlomen varnostni sistem postavili sami. Pri postavitvi protivlomnega varnostnega sistema bomo upoštevali vse zgoraj navedene zahteve.

3.3 Povzetek rešitve

Za uspešno izvedbo varnostnega sistema smo morali izbrati pravo strojno in programsko opremo. Na vhodna vrata in okna v vseh prostorih smo namestili brezžične magnetne kontakte, ki nam javijo, če so vrata oz. okna odprta ali zaprta, na prehod iz hodnika v kuhinjo z jedilnico in iz kuhinje z jedilnco v dnevno sobo pa smo namestili brezžični detektor gibanja. V dnevno sobo smo namestili tudi pametne žarnice, ki se bodo v primeru zaznanega gibanja samodejno prižgale. Kontrolne enote našega sistema smo namestili v posebno omarico v hodniku. Za spremljanje, upravljanje in avtomatizacijo našega sistema smo uporabili odprtokodno platformo Home Assistant (poglavje 4.4), za obveščanje in oddaljeno upravljanje s sistemom pa smo uporabili aplikacijo za nepsoredno sporočanje Telegram (poglavje 4.5).

V nadaljevanju diplomske naloge bomo podrobno opisali uporabljene pametne naprave, programsko opremo in nastavitve sistema. Na sliki 3.1 lahko vidite tloris stanovanja z vrisanimi pametnimi napravami.



Slika 3.1: Tloris stanovanja z vrisano strojno opremo

Poglavje 4

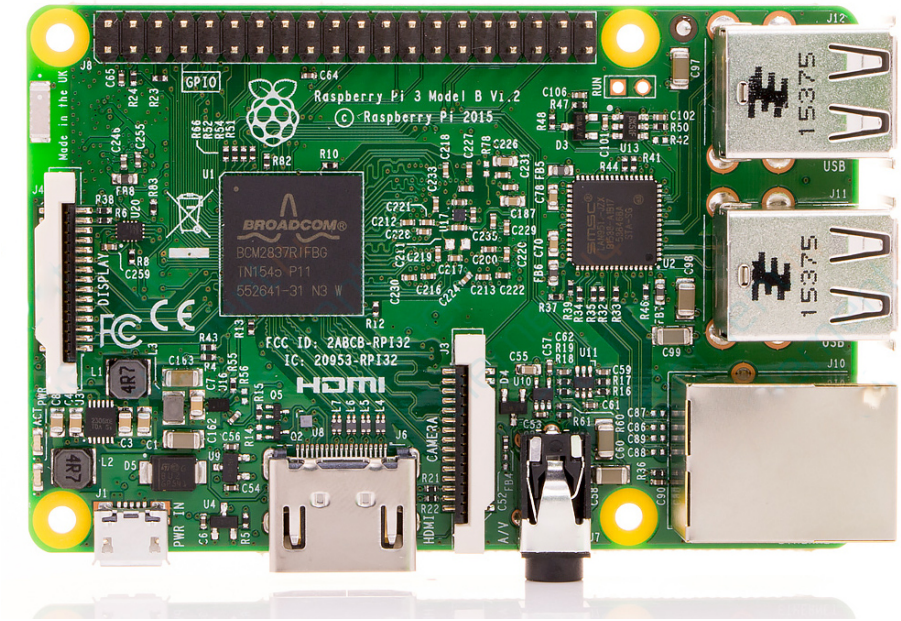
Strojna in programska oprema

4.1 Raspberry PI

Centralna kontrolna enota našega protivlomnega varnostnega sistema je mikroračunalnik Raspberry PI 3 Model B [22], ki ga bomo v nadaljevanju označevali z RPI. RPI smo izbrali na podlagi zahtev platforme Home Assitant, ki jo bomo uporabili za spremljanje, upravljanje in avtomatizacijo pametnih naprav v našem domu. Na RPI smo namestili operacijskim sistemom Rasbian z dodatkom Hassbian.

Raspbian operacijski sistem je odprtokoden in je narejen na Linuxovem jedru. Narejen je posebaj za Raspberry PI mikroračunalnike. Z dodatkom Hassbian pa omogoča podporo hišni optimizaciji, saj ima prednameščen sistem Home Assistant, ki smo ga opisali v poglavju 4.4.

Raspberry PI (slika 4.1) je zmogljiv mini računalnik, ki je bil ustvarjen v Združenem Kraljestvu z namenom ozaveščanja in širjenja znanja osnov računalniške pismenosti med otroci v šolah. Prva generacija teh mini računalnikov je izšla februrja 2012 (Raspberry PI 1 Model B), leta 2013 pa je izšel cenovno bolj ugoden a preprostejši Model A. Do danes je izšlo še nekaj izboljšanih modelov: PI 1 Model A+, PI 1 Model B+, PI 2 Model B, PI 3 Model B, PI Zero, PI Zero W in najnovejši PI 3 Model B+ [28]. Specifikacije naše centralne kontrolne enote so navedene v tabeli 4.1.



Slika 4.1: Raspberry PI 3 Model B [22]

Sistem na čipu	Broadcom BCM2837
Centralna procesna enota	Quad Core (4x ARM Cortex-A53), 1.2GHz, 64bit
Grafična procesna enota	Broadcom VideoCore IV
Delovni pomnilnik	1GB LPDDR2 (900 MHz)
Omrežje	10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth	Bluetooth 4.1 Classic, BLE
Hramba podatkov	microSD (32GB Sandisk ultra)
Ostala periferija	40-pin GPIO,HDMI, 4x USB 2.0., CSI, DSI, Ethernet, 3.5mm analogue audio-video jack

Tabela 4.1: Specifikacija Rasberry PI Model 3 B [22]

4.2 Senzorji Xiaomi Aqara

Aqara je vodilno kitajsko podjetje, ki ponuja cenovno ugodne, brezžične in energetsko varčne pametne naprave, s pomočjo katerih lahko naredimo naš dom pameten. Senzorji, ki jih trenutno ponujajo so brezžični magnetni kontakti, brezžični detektor gibanja, brezžični detektor iztekanja vode, brezžični merilec temperature in vlage, brezžični merilec treslajev in še mnogi drugi. Vse naprave je mogoče integrirati v večino odprtokodnih platform za upravljanje pametnih domov, lahko pa jih upravljamo tudi preko njihove uradne aplikacije Mi Home.

Za potrebe našega protivlomnega varnostnega sistema smo izbrali pet brezžičnih magnetnih kontaktov, ki smo jih namestili na vrata in okna in dva brezžična detektorja gibanaja (slika 4.2). Magnetne kontakte in detektorja gibanja smo preko pametnega zvezdišča (ang. smart hub) povezali s centralno kontrolno enoto.

Slika 4.2: Brezžični magnetni kontakt, pametno zvezdišče in brezžični detektor gibanja

4.2.1 Pametno zvezdišče

Pametno zvezdišče smo uporabili kot vmesni člen med centralno kontrolno enoto in Xiaomi Aqara senzorji. Poleg vloge vmesnega člena pa se pametno zvezdišče obnaša tudi kot sirena, ki se vklopi ob definiranem dogodku. Sirena

se vklopi, če je zaznano gibanje ali če je zaznano odpiranje vrat ali oken v času, ko je alarm vključen.

Pametno zvezdišče povezuje brezžične magnetne kontakte in brezžične detektorje gibanja s centralno kontrolno enoto. S senzorji, ki so povezani z zvezdiščem komunicira preko protokola ZigBee, s centralno kontrolno enoto pa preko brezžičnega omrežja (Wi-Fi). Zvezdišče lahko doseže vse naprave v radiju do 30 metrov [5]. Specifikacije so predstavljene v tabeli 4.2.

Napajanje	100V - 240V
Povezljivost	WiFi 2,4 GHz (802.11 b/g/n) in ZigBee
Dovoljena temperatura prostora	0C - 40C
Dovoljena vlažnost prostora	5% - 95%

Tabela 4.2: Specifikacije pametnega zvezdišča [5]

4.2.2 Brezžični magnetni kontakti

Brezžične magnetne kontakte smo namestili na protivlomna vhodna vrata in na okna v spalnici, kopalnici, kuhinji z jedilnico in dnevni sobi. Preko magnetnih kontaktov dobimo informacijo ali so vrata oz. okna odprta ali zaprta. Magnetni kontakti s pametnim zvezdiščom komunicirajo preko ZigBee protokola, napaja pa jih baterija CR1632, ki ima po specifikacijah (tabela 4.3) življensko dobo dolgo 2 leti [4].

Napajanje	baterija CR1632
Povezljivost	protokol ZigBee
Dovoljena temperatura prostora	-10C - 50C
Dovoljena vlažnost prostora	0% - 95 %
Največja razdalja med magneti	22 mm

Tabela 4.3: Specifikacije brezžičnega magnetnega kontakta [4]

4.2.3 Brezžični detektor gibanja

Brezžična detektorja gibanja smo namestili na prehodu iz hodnika v kuhinjo z jedilnico in iz kuhinje z jedilnico v dnevno sobo. Detektor gibanja je tako kot magnetni kontakti s pametnim zvezdiščom povezan preko brezžičnega protokola ZigBee. Detektor gibanja ima vgrajen IR senzor, ki zazna gibanje s spremljanjem razlike v toploti. Sposoben je zaznati gibanje na razdalji sedmih metrov in v območju 170 stopinj [6]. Specifikacije so navedene v tabeli 4.4.

Napajanje	baterija CR2450
Povezljivost	ZigBee
Dovoljena temperatura prostora	-10C - 45C
Največja razdalja	7m
Največje območje	170 stopinj

Tabela 4.4: Specifikacije brezžičnega detektorja gibanja [6]

4.3 Pametne žarnice

V dnevni sobi, ki gleda na glavno cesto, smo namesto navadnih žarnic namestili pametne LED žarnice. Pametne LED žarnice lahko oddaljeno prižigamo in ugašamo, omogočale pa nam bodo tudi nastavljanje urnika samodejnega prižiganja in ugašanja žarnic, kar bi lahko preprečilo morebiten vlom, saj bi vlomilcem dali vedeti, da stanovanje ni prazno.

Med pestro ponudbo pametnih žarnic, smo se odločili za Philipsov komplet Hue. Komplet sestavljata dve varčni 120 voltni LED žarnici s svetilnostjo 800 lumnov, ki jih je mogoče tudi zatemniti in vozlišče, ki povezuje žarnici s centralno kontrolno enoto. Vozlišče z žarnicama komunicira preko brezžičnega ZigBee protokola, z našo centralno kontrolno enoto pa komunicira preko HTTP protokola.

4.4 Home Assistant

Preden smo se odločili za platformo, ki nam bo omogočala lažji nadzor nad napravami v stanovanju smo se lotili primerjave odprtokodnih platform za spremljanje, upravljanje in avtomatizacijo pametnih naprav v našem domu. Med mnogimi rešitvami smo izbor platforme skrčili na tri kandidate (Home Assistant, openHab 2 in Eclipse Kapua), ki smo jih izbrali predvsem po tem kako je skupnost, ki razvija te platforme aktivna. Primerjali smo podatke iz platforme za gostovanje GitHub, ki ponuja gostovanje kode in statistiko udeležbe razvijalcev pri razvijanju produkta. Za obdobje od 5. januarja 2019 do 5. februarja 2019 smo primerjali število razvijalcev, število objav, število sprejetih in predlaganih sprememb ter število zaprtih in odprtih težav. Primerjava je predstavljena v tabeli 4.4.

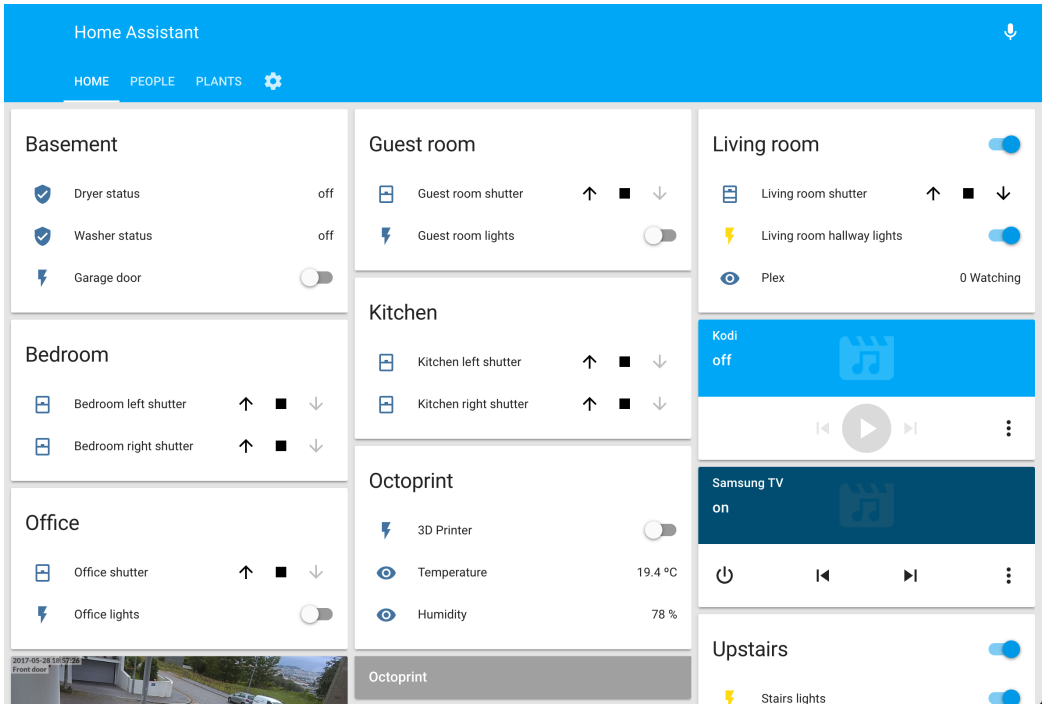
	Home Assistant	openHAB 2	Kapua
število razvijalcev	143	73	7
število objav	513	198	65
število sprejetih sprememb	416	197	45
število predlaganih sprememb	91	31	8
število zaprtih težav	364	57	49
število odprtih težav	235	64	17

Tabela 4.5: Primerjava platform za obdobje od 5.1. do 5.2. 2019

Ugotovili smo, da je med razvijalci najbolj priljubljena odprtokodna platforma Home Assistant, najmanj pa platforma Eclipse Kapua. Za platformo, ki bo nameščena na naš centralni kontrolni sistem in preko katere bomo upravljali naš protivlomen varnostni sistem smo izbrali Home Assistant

Home Assistant [13], v nadaljevanju HA, je odprtokodna platforma, ki omogoča spremljanje, upravljanje in avtomatizacijo vseh pametnih naprav v našem domu. Platforma je napisana v programskem jeziku Python3 in je zgrajena modularno, kar omogoča razvijalcu enostavno implementacijo podpore napravam, ki še niso podprte.

Na naš centralni kontrolni sistem smo namestili operacijski sistem Raspbian in dodatek Hassbian, ki podpira HA. Preden lahko začnemo uporabljati HA ga moramo nastaviti po naših željah. Nekatere osnovne nastavitve lahko spremenimo preko uporabniškega vmesnika ostale pa moramo spremeniti neposredno v nastavitveni datoteki. Kako so nastavitvene datoteke organizirane in kako smo mi nastavili sistem bomo podrobno opisali v poglavju 5.



Slika 4.3: Vmesnik Home Assistant [13]

4.5 Telegram

Telegram [23] je oblačna storitev namenjena pošiljanju sporočil in klicanju med uporabniki. Uporabniku ponuja izdelavo lastnega robota (ang. bot), ki se odziva na sporočila, omembe in ga je možno integrirati tudi v druge programe. Našega centralnega sistema ne želimo direktno izpostaviti zunanjemu svetu, zato bomo uporabili Telegram, preko katerega bomo komunicirali s platformo HA. Nastavili bomo robota (poglavje 5.4), ki bo podpiral ukaze za poizvedovanje in izvajanje procedur preko platforme HA:

- Ukaz *status_hodnik_vrata* sproži poizvedbo in odgovori z *odprta*, če so vrata v hodniku odprta ali z *zaprta*, če so vrata v hodniku zaprta.
- Ukaz *status_spalnica_okno* sproži poizvedbo in odgovori z *odprto*, če je okno v spalnici odprto ali z *zaprto*, če je okno v spalnici zaprto.
- Ukaz *status_kopalnica_okno* sproži poizvedbo in odgovori z *odprto*, če je okno v kopalnici odprto ali z *zaprto*, če je okno v kopalnici zaprto.
- Ukaz *status_kuhinja_okno* sproži poizvedbo in odgovori z *odprto*, če je okno v kuhinji odprto ali z *zaprto*, če je okno v kuhinji zaprto
- Ukaz *status_dsoba_okno* sproži poizvedbo in odgovori z *odprto*, če je okno v dnevni sobi odprto ali z *zaprto*, če je okno v dnevni sobi zaprto.
- Ukaz *status_dsoba_luc* sproži poizvedbo in odgovori z *vklopljena*, če je luč v dnevni sobi vklopljena ali z *izklopljena*, če je luč v dnevni sobi izklopljena.
- Ukaz *akcija_dsoba_luc_vklopi* vklopi luč v dnevni sobi.
- Ukaz *akcija_dsoba_luc_izklopi* izklop luč v dnevni sobi.
- Ukaz *akcija_alarm_vklopi* vklopi alarm.
- Ukaz *akcija_alarm_izklopi* izklopi alarm.

Poglavje 5

Namestitev in nastavitev sistema

Na mikro računalnik RPI smo namestili odprtokodno platformo Home Assistant, ki nam omogoča spremljanje, upravljanje in avtomatizacijo pametnih naprav v našem stanovanju. Preko spletne strani smo prenesli sliko sistema Hasbian na katerem je že prednameščen Home Assistant. Sliko sistema smo z odprtokodnim programom Etcher namestili na 32 GB mikro SD pomnilno kartico, ki služi tudi kot spominski medij. Po namestitvi slike sistema, smo spominsko kartico vstavili v RPI in po približno desetith minutah je vsa nastavitev in namestitev sistema končana. Po prvem zagonu moramo počakati še približno deset minut, da se namesti najnovejša verzija sistema Home Assistant in nato lahko do njega dostopa preko ukazne vrstice ali pa kar preko brskalnika na naslovu http://hassbian.local:8123. Podroben opis namestitve sistema si lahko preberete na uradni spletni strani platforme HA [13].

Sistem nastavljamo preko nastavitvenih datotek, ki se nahajajo v domačem direktoriju uporabnika homeassistant. Do tega direktorija lahko dostopamo preko ssh povezave s privzetim uporabnikom pi in geslom raspberry. Seveda je priporočljivo da z ukazom passwd spremenimo geslo ali pa da namesto z geslom do RPI dostopamo s privatnim ključem.

5.1 Organizacija nastavitvenih datotek

Nastavitve platforme HA, ki jih uporabnik lahko spreminja, so shranjene v nastavitveni datoteki *configuration.yaml*. Datoteka se ob prvem zagonu HA samodejno zgeneira, v njej pa so definirane vse komponente, ki se naložijo ob zagonu platforme HA.

Nastavitve so v datoteki zapisane v formatu YAML. YAML je uporabniku prijazen format za serializacijo podatkov in je zasnovan tako, da uporabniku olajša delo s podatki. YAML je prenosljiv med programskimi jeziki, je ekspresiven in razširljiv in je lahek za implementacijo [16].

Z dodajanjem komponent v platformo HA, postane nastavitvena datoteka *configuration.yaml* nepregledna. HA nam omogoča, da nastavitveno datoteko razdelimo na več krajših in bolj obvladljivih datotek. Pri tem moramo paziti na zamike in presledke, ki so predpisani za YAML format. Nastavitvena datoteka je shranjena v čistem textu (ang. plain text) zato je priporočljivo, da vsa gesla in pomembne podatke, kot so API ključi shranimo v posebno datoteko *secrets.yaml*. Tudi v našem primeru smo nastavitveno datoteko razdelili na več manjših delov, ki so opisani v spodnjem seznamu in gesla ter pomembne podatke združili v posebno datoteko. V spodnjem seznamu so opisane vse nastavitvene datoteke, ki smo jih uporabili:

- V nastavitveni datoteki *alarm.yaml* so zapisane nastavitve alarma in procedur za vklop in izklop alarma.
- V nastavitveni datoteki *automations.yaml* so zapisane nastavitve vseh procedur, ki se zaženejo ob določenem dogodku. Procedure smo definirali za dogodke kot so: platforma HA prejeme sporočilo preko Telegrama, brezžični detektor gibanja zazna gibanje, brezžični magnetni kontakt zazna odprtje vrat ali oken.
- V nastavitveni datoteki *sensors.yaml* so zapisane nastavitve pametnih naprav. V našem primeru so to brezžični detektorji gibanja, brezžični magnetni kontakti in pametne žarnice.

- V nastavitveni datoteki *groups.yaml* so zapisane nastavitve združevanja komponent po funkcijah. Mi smo združili dve žarnici v eno luč.
- V nastavitveni datoteki *notifications.yaml* so zapisane nastavitve naprav na katere bo platforma HA pošiljala obvestila.
- V nastavitveni datoteki *secrets.yaml* so zapisana vsa gesla in občutljivi podatki.
- V nastavitveni datoteki *devices.yaml* so zapisane vse naprave z MAC naslovi, ki so prijavljene v lokalno omrežje.

```
homeassistant:
  name: Pameten protivlomen sistem
  unit_system: metric
  time_zone: Europe/Ljubljana
  latitude: !secret home_latitude
  longitude: !secret home_longitude

  alarm_control_panel: !include alarm.yaml
  automation: !include automations.yaml
  sensor: !include sensors.yaml
  group: !include groups.yaml
  notify: !include notifications.yaml
  device_tracker: !include devices.yaml
```

5.2 Nastavitev senzorjev Xiaomi Aqara

Platforma HA nam omogoča enostavno integracijo pametnih naprav proizvajalca Xiaomi Aqara. V našem primeru bomo s platformo povezali pametno vozlišče, brezžične magnetne kontakte in brezžične detektorje gibanja.

Najprej smo morali s platformo povezati pametno vozlišče. V platformo lahko integriramo eno ali več pametnih vozlišč. To naredimo tako, da v glavno nastavitveno datoteko *configuration.yaml* vnesemo ključ vozlišča, ki ga povezujemo. Ključ pridobimo preko uradne mobilne aplikacije MiHome [18], ki si jo lahko na naš pametni telefon prenesmo iz trgovine Google play oziroma iz trgoivne App Store.

```
xiaomi_aqara:
  # HA se bo petkrat poskusal povezati s pametnim vozliščem
  discovery_retry: 5
  gateways:
    - key: !secret xiaomi_aqara_key
```

Sedaj, ko je pametno vozlišče integrirano s platformo HA, lahko povežemo tudi brezžične magnete kontakte in brezžične detektorje gibanja. S pametnim zvezdiščem jih povežemo preko uradne aplikacije MiHome nameščene na našem pametnem telefonu. Preko aplikacije pridobimo tudi edinstveni identifikator naprave, ki ga uporabimo pri integraciji senzorjev s HA. V tabeli 5.1 je prikazano kako so brezžčni magnetni kontaktki in brezžčni detekorji gibanja predstavljeni v platformi HA.

Senzor	Tip	Vrednost	Dogodek
Brezžični detektor gibanja	binarni	on,off	motion
Brezžični magnetni kontakti	binarni	on,off	-

Tabela 5.1: Predstavitev senzorjev v platformi HA

V nadaljevanju bomo predstavili integracijo magnetnih kontakov in detektorja gibanja s platformo HA.

Nastavitev brezžičnega detektorja gibanja

Brezžični detektor gibanja je binarni senzor, ki ob zaznanem gibanju sproži dogodek z imenom *motion* in stanje senzorja spremeni iz 0 (*off*) v 1 (*on*). De-

tektor gibanja v dnevni sobi s platformo HA integriramo tako, da v datoteko *sensors.yaml* vpišemo spodnje nastavitve, ki nastavijo odziv ob zaznanem gibanju. Postopek ponovimo tudi za detektor gibanja, ki se nahaja v kuhinji.

```
# nastavitev besedila in ikone za detektor gibanja v
# dnevni sobi
  - platform: template
    sensors:
      living_room_movement:
        # nastavi primerno besedilo
        value_template: '
          {%if is_state(
          "binary_sensor.motion_sensor_158d0001a66256",
          "on")%}
             Gibanje je zaznano
          {% else %} Ni gibanja {% endif %}'
        # nastavi primerno ikono
        icon_template: '
          {%if is_state(
          "binary_sensor.motion_sensor_158d0001a66256",
          "on")%}
              mdi:run-fast
          {% else %} mdi:sleep {% endif %}'
```

Besedilo in ikono, ki se spreminjata glede na vrednost, ki jo senzor sporoča nastavimo tako, da definiramo predlogo (ang. template), ki bo glede na vrednost iz brezžičnega detektorja gibanja, uporabniku prikazala primerno besedilo z ikono. V primeru zaznanega gibanja se bo izpisalo besedilo *Gibanje je zaznano*, če gibanje ne bo zaznano pa besedilo *Ni gibanja*. Poleg primernega besedila, se bo prikazala tudi primerna ikona.

V primeru zaznanega gibanja želimo tudi, da nas platforma HA o tem obvesti preko platforme Telegram, ki smo jo opisali v poglavju 4.5. Tako kot

vse senzorje moramo tudi Telegram integrirati in nastaviti v platformi HA (postopek je opisan v poglavju 5.4). V spodnjih nastavitvah, ki se nahajajo v datoteki *automations.yaml*, je nastavljeno, da se v primeru zaznanega gibanja v dnevni sobi preko Telegrama pošlje sporočilo o zaznanem gibanju.

```
# nastavitev posiljanja obvestil preko platforme Telegram
- alias: lr_motion_detected
  trigger:
    - platform: state
      entity_id: binary_sensor.motion_sensor_158d0001a66256
      from: 'off'
      to: 'on'
  action:
    - service: notify.miha_tgram
      data:
        title: 'Alarm: Zaznano gibanje!'
        message: 'Zaznano gibanje v dnevni sobi!'
```

Nastavitev brezžičnega magnetnega kontakta

Na podoben način kot smo integrirali in nastavili brezžični detektor gibanja v HA, bomo integrirali in nastavili tudi brezžični magnetni kontakt. Magnetni kontakti so v platformi HA predstavljeni kot binarni senzor in so lahko v stanju 0 (*off*), vrata so zaprta, oziroma 1 (*on*), vrata so odprta. V nadaljevanju si bomo pogledali nastavitve brezžičnega magnetnega kontakta na vhodnih vratih. Nastavitve za magnetni kontakt na vratih se skoraj ne razlikujejo od nastavitev magnetnih kontaktov v spalnici, kopalnici, kuhinji z jedilnico in dnevni sobi.

```
# nastavitev besedila in ikone za magnetni kontakt na
# vhodnih vratih
sensor:
  - platform: template
```

```
  sensors:
    living_room_movement:
      # nastavi primerno besedilo
      value_template: '
        {%if is_state(
        "binary_sensor.status158d0001a77777",
        "on")%}
           Vrata so odprta
        {% else %} Vrata so zaprta {% endif %}'
      # nastavi primerno ikono
      icon_template: '
        {%if is_state(
        "binary_sensor.status158d0001a77777",
        "on")%}
            mdi:door-open
        {% else %} mdi:door-closed {% endif %}'
```

```
# nastavitev posiljanja obvestil preko platforme Telegram
- alias: main_door_open
  trigger:
    - platform: state
      entity_id: binary_sensor.status158d0001a77777
      from: 'off'
      to: 'on'
  action:
    - service: notify.miha_tgram
      data:
        title: 'Alarm: Vrata so odprta!'
        message: 'Vrata so odprta!'
```

5.3 Nastavitev žarnic Philips Hue

V dnevno sobo smo namestili žarnice Philips Hue, ki so povezane z zvezdiščem Hue Bridge. Žarnici smo povezali s platformo HA in nastavili, da se bosta samodejno prižgali, če bo v stanovanju zaznano gibanje oziroma, če bo zaznano odprtje vhodnih vrat ali pa katerega izmed oken.

Priporočljivo je, da integriramo Philips Hue platformo v HA platformo s pomočjo "discovery"komponente, ki nam jo ponuja HA. Lahko pa jo integriramo tudi sami. To naredimo tako, da v nastavitveno datoteko *configuration.yaml* vpišemo spodnje nastavitve.

```
hue:
  bridges:
    # IP naslov na katerem je dosegljiv Hue Bridge
    - host: !secret hue_bridge_host
```

Hue omogoča nastavljanje različnih skupin žarnic, tako da lahko naprimer prižigamo vse žarnice v dnevni sobi naenkrat ali pa samo določene žarnice. To lahko nastavimo preko njihove uradne aplikacije ali pa preko končnih točk, ki jih definira njihov API. V našem primeru imamo obe žarnici v eni luči, zato bomo v nastavitveni datoteki *groups.yaml* definirali skupino *dnevna_soba_luc* in ju dodali vanjo.

5.4 Podpora ukazom preko Telegram robota

V podpoglavju 4.5 smo opisali platformo Telegram, ki omogoča izdelavo in nastavitev lastnega robota (ang. bot). Robota bomo nastavili tako, da bo znal izvesti ukaze glede na našo zahtevo. Z uporabo robota smo zagotovili določeno varnost našega sistema, saj ga nismo direktno izpostavili svetu.

Robota smo ustvarili s pomočjo že integriranega robota BotFather [7]. V aplikaciji Telegram ustvarimo nov pogovor v katerega vpišemo ukaz *newbot* in vnesemo ime robota in uporabniško ime preko katerega ga bomo lahko

referencirali. BotFather nato generira žeton, ki ga bomo skupaj z identifikatorjem klepeta (*chat_id*), preko katerega povemo robotu s katerimi uporabniki lahko komunicira, uporabili v nastavitvah v našem sistemu. V nastavitveno datoteko *configuration.yaml* zapišemo naslednje nastavitve.

```
telegram_bot:
  - platform: polling
    api_key: !secret telegram_api_key
    allowed_chat_ids:
      - !secret allowed_chat_id
```

Naslednji korak je nastavitev podprtih ukazov, ki smo jih predstavili v poglavju 4.5. V nastavitveni datoteki *automations.yaml* smo nastavili odgovor, ki ga bo naš sistem poslal ob prejemu ukaza *status_dsoba_luc*. Proceduro smo poimenovali *telegram_status_dsoba_luc* in se bo sprožila ob prejemu ukaza *status_dsoba_luc*. Procedura pošiljatelju odgovori s stanjem luči v dnevni sobi in ponudi možnost vklopa ali izklopa luči.

```
  - alias: telegram_status_dsoba_luc
    initial_state: on
    trigger:
      platform: event
      event_type: telegram_command
      event_data:
        command: '/status_dsoba_luc'
    action:
      service: telegram_bot.send_message
      data_template:
        target: "{{ trigger.event.data.user_id }}"
        message: "
          {% if states.light.ds1.state == "on"
             && states.light.ds2.state == "on"}
```

```
        Vklopljena
      {% else %}
        Izklopljena
      {% endif %}
    keyboard: [
      "Vklopi:/akcija_dsoba_luc_vklopi",
      "Izklopi:/akcija_dsoba_luc_izklopi"
    ]
```

V primeru, da uporabnik vklopi luči se izvede procedura *akcija_dsoba_luc_vklopi*, ki prižge luč v dnevni sobi in odgovori uporabniku z odgovorom *Vklopljena*.

```
- alias: telegram_akcija_ds_luc_vklopi
  initial_state: on
  trigger:
    platform: event
    event_type: telegram_command
    event_data:
      command: '/akcija_ds_luc_vklopi'
  action:
    - service: homeassistant.turn_on
      entity_id: group.lr_lights
    - service: telegram_bot.send_message
      data_template:
        target: "{{ trigger.event.data.user_id }}"
        message: "Vklopljena"
```

5.5 Nastavitev alarma

Glavne tri zahteve našega protivlomnega sistema so, da se vklopi ob točno določenem času, izklopi ob točno določenemu času in da se vklopi, če zazna da nikogar ni doma. Platforma HA omogoča nastavitev komponente imenovane *alarm_control_panel*. Komponento, tako kot vse ostale, omogočimo v nastavitveni datoteki *configuration.yaml*. Poleg mnogih nastavitev ji lahko določimo skrivno kodo (PIN), ki jo zaradi varnosti nastavimo v datoteki *secrets.yaml* in jo uporabimo pri vklopu in izklopu alarma. Alarm se lahko nahaja v več stanjih, mi bomo za naše potrebe uporabili dva:

- Stanje alarma *armed_away*, ki označuje, da je alarm vklopljen.
- Stanje alarma *disarmed*, ki označuje, da je alarm izklopljen.

Spodnje nastavitve so shranjene v datoteki *alarm.yaml* in nastavijo proceduro za vklop alarma. Procedura se sproži, če naprava z identifikatorjem *device_tracker.miha_iphone* 15 minut ni prijavljen v lokalno omrežje in če velja pogoj, da je alarm v stanju *disarmed*. Procedura postavi alarm v stanje *armed_away*.

```
- alias: arm_alarm
  initial_state: 'off'
  trigger:
    - platform: state
      entity_id: device_tracker.miha_iphone
      to: 'not_home'
      for:
        minutes: 15
  condition:
    condition: state
    entity_id: alarm_control_panel.alarm
    state: 'disarmed'
  action:
```

```
    service: alarm_control_panel.alarm_arm_away
    data:
      entity_id: 'alarm_control_panel.alarm'
      code: !secret alarm_code
```

Nasprotno od procedure *arm_alarm*, procedura *disarm_alarm* izklopi alarm. Procedura se sproži, ko se naprava z identifikatorjem *device_tracker.miha_iphone* prijavi v omrežje in če velja pogoj, da je bil alarm prej v stanju *armed_away*. Procedura postavi alarm v stanje *disarmed*.

```
- alias: disarm_alarm
  trigger:
    - platform: state
      entity_id: device_tracker.miha_iphone
      to: 'home'
  condition:
    condition: state
    entity_id: alarm_control_panel.alarm
    state: 'armed_away'
  action:
    service: alarm_control_panel.alarm_disarm
    data:
      entity_id: 'alarm_control_panel.alarm'
      code: !secret alarm_code
```

Poglavje 6

Varnost sistema

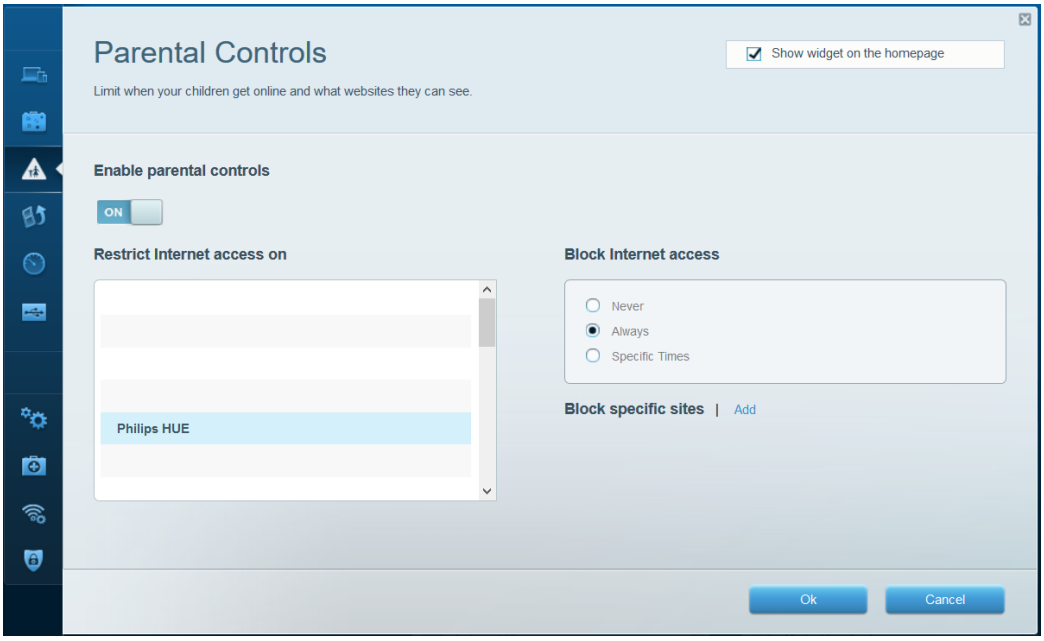
Med načrtovanjem in postavljanjem protivlomnenga varnostnega sistema se nam je porajalo vprašanje kako bomo do sistema dostopali iz zunanjega omrežja in kako bomo sistem zavarovali pred morebitnimi vdori. Zaradi dejstva, da smo uporabili zelo poceni senzorje in ostale komponente nas je zanimalo tudi, če so zaradi tega naši podatki dostopni proizvajalcu.

Po postavitvi sistema smo z odprtokodnim programom Wireshark [31] analizirali omrežni promet in ugotovili, da Xiaomi pametno zvezdišče komunicira s strežniki na Kitajskem, zvezdišče na katerega se povezujejo pametne žarnice pa z Amazonovimi spletnimi servisi. Seveda ne želimo, da je naša zasebnost tako izpostavljena zato smo najprej blokirali ves promet v internet in dovolili samo lokalno komunikacijo med napravami. Vso zunanjo komunikacijo smo onemogočili kar na usmerjevalniku, ki nam omogoča, da preko starševske kontrole za določeno napravo na statičnem IP naslovu blokiramo dostop do interneta. Slika 6.1 prikazuje vmesnik na usmerjevalniku za blokado dostopa do interneta.

Dostop do uporabniškega vmesnika platforme HA smo omogočili samo v lokalnem omrežju. V primeru, da bi želeli do nje dostopati iz zunanjega omrežja, pa smo to omogočili preko SSH tunela. Kaj je SSH tunel in kako smo ga nastavili si lahko preberete v podpoglavju 6.1.

HA nam vsa obvestila in opozorila pošilja na platormo Telegram. Preko te

platforme pa lahko tudi poizvedujemo po stanju naprav v našem stanovanju in izvajamo preddefinirane akcije. Celotna komunikacija med HA in Telegramom je v celoti šifrirana, Telegram pa ponuja celo visoko denarno napravo tistemu, ki bi uspel dešifrirati sporočilo poslano preko njihove platforme.



Slika 6.1: Nastavitev onemogočenega dostopa do interneta za določeno napravo v omrežju

6.1 Nastavitev SSH tunela

SSH (ang. Secure Shell) je šifrirni omrežni protokol, ki omogoča prijavo na oddaljen sistem in izvajanje programov. SSH omogoča tudi tuneliranje, posredovanje vrat (ang. port forwarding) in prenos datotek [19].

Za vzpostavitev tunela smo morali najprej omogočiti posredovanje vrat iz vrat številka 22 na vrata na katerih teče HA v našem lokalnem omrežju. To smo storili preko uporabniškega vmesnika na našem usmerjevalniku. Nastavili smo tudi DNS naslov preko odprtokodnega servisa DuckDNS [9], ki

omogoča dostop do centralnega sistema kljub temu, da imamo dinamičen IP, ki se redno spreminja. Spodnji ukaz prikazuje vzpostavitev SSH tunela.

```
ssh -L 8444:localhost:8123 uporabnik@10.42.0.1
```

Ukaz lahko preberemo kot:

- Vzpostavili bomo SSH tunel s posredovanjem lokalnih vrat (**ssh -L**).
- Do sistema bomo dostopali preko lokalnih vrat številka **8444**.
- Uporabniški vmesnik HA bomo prikazali na naslovu **localhost**.
- Oddaljeni uporabniški vmesnik HA se nahaja na vratih številka **8123**.
- Uporabniško ime oddaljenega sistema je **uporabnik**.
- Naslove oddaljenega sistema je **10.42.0.1**.

Poglavje 7

Testiranje sistema

Testiranje je del razvoja in postavitve vsakega informacijskega sistema. Testiranje informacijskega sistema je potrebno zaradi potrditve, da sistem deluje v skladu s specifikacijami in zahtevami. Zahteve za naš protivlomen varnostni sistem smo postavili v poglavju 3.2.

Poznamo ročno in avtomatsko testiranje, ki se izvaja s pomočjo testnih orodji. Ročno testiranje je testiranje kjer se tester postavi v vlogo končnega uporabnika in po predpisanih testnih primerih testira delovanje sistema [17]. V našem protivlomnem varnostnem sistemu smo uporabili šest različnih komponent:

- brezžični detektor gibanja,
- brezžični magnetni kontakt (vrata in okna),
- sirena (pametno zvezdišče Xiaomi),
- pametne žarnice.

Določili smo tri scenarije (A, B, C) v katerih smo testirali vsako od komponent. Scenarij A definira vključen sistem med osmo uro zjutraj in tretjo popoldan, scenarij B definira izključen sistem med tretjo uro popoldan in osmo uro zjutraj naslednjega dne, scenarij C definira zaznavo nepovezanega

telefona v omrežje in posledično samodejno vključitev sistema. Definirani scenariji:

(A) Sistem se samodejno vključi in je vključen od 8:00 do 15:00.

(B) Sistem se samodejno izključi in je izključen od 15:00 do 8:00.

(C) Sistem zazna, da telefon ni povezan v omrežje in se samodejno vključi.

Vsak scenarij definira dve možni akciji (*telegram* in *sirena*). Akcija *telegram* definira pošiljanje obvestila uporabniku preko platforme Telegram, akcija *sirena* pa definira vklop sirene:

- Akcija *telegram*, označuje akcijo, ki uporabniku pošlje obvestilo preko platforme Telegram.
- Akcija *sirena*, označuje akcijo, ki vklopi sireno.

V tabeli 7 je definiranih vseh 18 testnih primerov s pričakovanimi in dejanskimi rezultati. Vsi dejanski testni rezultati se ujemajo s pričakovanimi zato lahko rečemo, da naš protivlomen varnostni sistem deluje po specifikacijah in zajema vse naše zahteve.

#	Testni primer	Pričakovan odziv		Dejanski odziv	
		Telegram	Sirena	Telegram	Sirena
1	A - odprto okno v spalnici	DA	DA	DA	DA
2	B - odprto okno v spalnici	NE	NE	NE	NE
3	C - odprto okno v spalnici	DA	NE	DA	NE
4	A - odprto okno v kopalnici	DA	DA	DA	DA
5	B - odprto okno v kopalnici	NE	NE	NE	NE
6	C - odprto okno v kopalnici	DA	NE	DA	NE
7	A - odprto okno v kuhinji	DA	DA	DA	DA
8	B - odprto okno v kuhinji	NE	NE	NE	NE
9	C - odprto okno v kuhinji	DA	NE	DA	NE
10	A - odprto okno v dnevni sobi	DA	DA	DA	DA
11	B - odprto okno v dnevni sobi	NE	NE	NE	NE
12	C - odprto okno v dnevni sobi	DA	NE	DA	NE
13	A - zaznano gibanje v hodniku	DA	DA	DA	DA
14	B - zaznano gibanje v hodniku	NE	NE	NE	NE
15	C -zaznano gibanje v hodniku	DA	NE	DA	NE
16	A - zaznano gibanje v dnevni sobi	DA	DA	DA	DA
17	B - zaznano gibanje v dnevni sobi	NE	NE	NE	NE
18	C - zaznano gibanje v dnevni sobi	DA	NE	DA	NE

Tabela 7.1: Testni primeri s pričakovanimi in dejanskimi rezultati

Poglavje 8

Zaključek

V diplomskem delu smo predstavili postavitev lastnega pametnega brezžičnega protivlomnega varnostnega sistema. V uvodnem delu diplomskega dela smo opisali internet stvari in predstavili poenostavljeno arhitekturo, ki zajema fizično, omrežno in aplikacijsko plast. V osrednjem delu diplomskega dela smo opisali problem, ki ga rešujemo in zakaj smo se odločili za postavitev lastnega protivlomnega sistema. Predstavili smo strojno in programsko opremo, ki smo jo uporabili v naši reštivi in kako smo protivlomen sistem namestili in nastavili. Posebno poglavje pa smo namenili tudi varnostni in testiranju sistema.

8.1 Sklepne ugotovitve

Postavili smo delujoč protivlomen varnostni sistem, ki ustreza našim željam in zahtevam. Kljub temu smo ugotovili, da ima sistem kritično pomanklji-vosti. Protivlomen varnostni sistem v primeru izpada elektirčne energije preneha delovati. To bomo rešili tako, da bomo poskrbeli za dodatno napajanje. Namestili bomo dodaten vir napajanja (akumulator), ki bo poskrbel za delovanje sistema v primeru izpada glavnega vira. Protivlomen varnostni sistem smo povezali s platformo HA, ki omogoča tudi dodajanje mnogih drugih pametnih naprav. V podpoglavju 8.2 smo opisali možnosti nadgradnje

našega doma.

8.2 Možnosti nadgradnje

Poleg protivlomnega varnostnega sistema imamo s platformo HA povezan tudi pametni zvočnik Echo Dot, ki ga izdeluje podjetje Amazon. Na zvočniku je nameščen pametni osebni asistent Alexa in nam omogoča glasovno upravljanje naprav, ki jih imamo nameščene v pametnem domu.

Pametni osebni asistent Alexa, je sposoben dvosmerne komunikacije in se odziva na glasovne ukaze. Glasovno prepoznavanje ukazov je omejeno na angleški, nemški in japonski jezik. Preden izvedemo ukaz, moramo pametni zvočnik "prebuditi", to naredimo z besedo "Alexa".

Preko pametnega zvočnika, ki ga imamo nameščenga v dnevni sobi, kjer preživimo največ časa, lahko trenutno upravljamo (vklapljamo in izklapljamo) luč v dnevni sobi. V prihodnosti nameravamo vse električne vtičnice zamenjati s pametnimi električnimi vtičnicami in jih povezati s HA. To nam bo omogočilo upravljanje naprav, ki se niso sposobne same povezati s HA. Zamenjali bomo tudi vse navadne LED žarnice s pametnimi, da jih bomo lahko upravljali preko platforme HA oz. preko pametnega zvočnika.

Literatura

[1] Mohammed ABDMEZIEM, D Tandjaoui, and Imed Romdhani. *Architecting the Internet of Things: State of the Art*, pages 55–75. 08 2015.

[2] HIGHEST AND LOWEST STATES AT RISK FOR BURGLARY. Dosegljivo: `https://www.adt.com/burglary-odds-across-america`. [Dostopano: 30. 1. 2019].

[3] Slavko Amon and UL Založba. Senzorji in aktuatorji. *Fakulteta za elektrotehniko, Dosegljivo: http://lms. fe. uni-lj. si/amon/knjiga/Senzorji_in_aktuatorji. pdf [Dostopano: 3. 5. 2016]*, 2013.

[4] Aqara Door and Window Sensor. Dosegljivo: `https://www.aqara.com/en/door_and_window_sensor-product.html`. [Dostopano: 13. 2. 2019].

[5] Aqara Hub. Dosegljivo: `https://www.aqara.com/en/smart_hub-product.html`. [Dostopano: 10. 2. 2019].

[6] Aqara Motion Sensor. Dosegljivo: `https://www.aqara.com/en/motion_sensor.html`. [Dostopano: 13. 2. 2019].

[7] Bots: An introduction for developers. Dosegljivo: `https://core.telegram.org/bots`. [Dostopano: 13. 2. 2019].

[8] CMU SCS Coke Machine. Dosegljivo: `https://www.cs.cmu.edu/~coke/`. [Dostopano: 8. 1. 2019].

[9] Duck DNS. Dosegljivo: https://www.duckdns.org/. [Dostopano: 13. 2. 2019].

[10] IoT Marches Into the Enterprise, Transformation Follows Quickly. Dosegljivo: http://info.forbes.com/rs/790-SNV-353/images/Intel-IoT%231-REPORT-FINAL-WEB.pdf. [Dostopano: 30. 1. 2019].

[11] Kirsten Gram-Hanssen and Sarah J Darby. "home is where the smart is"? evaluating smart home research and approaches against the concept of home. *Energy Research & Social Science*, 37:94–101, 2018.

[12] David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, and Jerome Henry. *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*. Cisco Press, 2017.

[13] Home Assitant. Dosegljivo: https://www.home-assistant.io. [Dostopano: 10. 2. 2019].

[14] Internet of Things Done Wrong Stifles Innovation. Dosegljivo: https://www.informationweek.com/strategic-cio/executive-insights-and-innovation/internet-of-things-done-wrong-stifles-innovation/a/d-id/1279157. [Dostopano: 8. 1. 2019].

[15] State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating. Dosegljivo: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b. [Dostopano: 9. 1. 2019].

[16] D Levanon, V Negreanu, Y Bernstein, I Bar-Am, L Avivi, and YAML Groner. Aml1, aml2, and aml3, the human members of the runt domain gene-family: cdna structure, expression, and chromosomal localization. *Genomics*, 23(2):425–432, 1994.

[17] Manual Testing Tutorial for Beginners: Concepts, Types, Tool . Dosegljivo: https://www.guru99.com/manual-testing.html. [Dostopano: 13. 2. 2019].

[18] Mi Home - xiaomi smarthome. Dosegljivo: https://itunes.apple.com/us/app/mi-home-xiaomi-smarthome/id957323480?mt=8. [Dostopano: 13. 2. 2019].

[19] OpenSSH. Dosegljivo: https://www.openssh.com/. [Dostopano: 13. 2. 2019].

[20] Keyur K Patel, Sunil M Patel, et al. Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5), 2016.

[21] Letna poročila o delu policije. Dosegljivo: https://www.policija.si/o-slovenski-policiji/statistika. [Dostopano: 13. 2. 2019].

[22] Raspberry Pi. Dosegljivo: https://www.raspberrypi.org/. [Dostopano: 13. 2. 2019].

[23] Telegram. Dosegljivo: https://telegram.org/. [Dostopano: 13. 2. 2019].

[24] The Internet of Things Reference Model. Dosegljivo: http://cdn.iotwf.com/resources/71/IoT_Reference_Model_White_Paper_June_4_2014.pdf. [Dostopano: 8. 1. 2019].

[25] Bluetooth. Dosegljivo: https://en.wikipedia.org/wiki/Bluetooth. [Dostopano: 13. 2. 2019].

[26] Communication protocol. Dosegljivo: https://en.wikipedia.org/wiki/Communication_protocol. [Dostopano: 3. 2. 2019].

[27] Hypertext Transfer Protocol. Dosegljivo: https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol. [Dostopano: 3. 2. 2019].

[28] Raspberry Pi. Dosegljivo: https://en.wikipedia.org/wiki/Raspberry_Pi. [Dostopano: 10. 2. 2019].

[29] Wi-Fi. Dosegljivo: https://en.wikipedia.org/wiki/Wi-Fi. [Dostopano: 13. 2. 2019].

[30] Zigbee. Dosegljivo: https://en.wikipedia.org/wiki/Zigbee. [Dostopano: 13. 2. 2019].

[31] Wireshark. Dosegljivo: https://www.wireshark.org/. [Dostopano: 13. 2. 2019].

[32] Ning Ye, Yan Zhu, Ru-chuan Wang, Reza Malekian, and Lin Qiaomin. An efficient authentication and access control scheme for perception layer of internet of things. *Applied Mathematics & Information Sciences*, 8(4):1617, 2014.

[33] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.